Stiff Equations and Absolute Stability

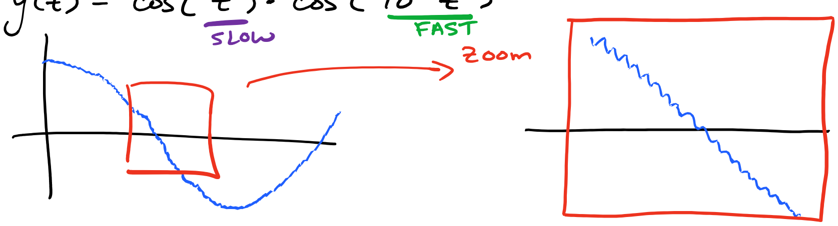
Thursday, November 19, 2020 9:23 PM

Ch. 5.11 is "Stiff Differential Eq'n" but it's really also about stability of a new kind

Stiff Equations

There's not a precise definition of a stiff equation, and it's really a scale, just like how ill-conditioning is a scale. The basic phenomenon is when our solution $y(t)$ (to the ODE) has very fast and very slow timescales.

Ex. $y(t) = \cos(\underbrace{t}_{\text{SLOW}}) \cdot \cos(\underbrace{10^4 t}_{\text{FAST}})$



① Slow timescales

These are not problematic by themselves, but the issue is that it means we usually want a large T

IVP: $y' = f(t,y)$
$y(0) = y_0$
$0 \le t \le T$

② Fast timescales

Ex. $\cos(10^4 \cdot t)$ or $e^{-100 \cdot t}$

If we took a large stepsize h, we'd jump over these features and not resolve them at all!

The issue: large T, but due to fast behaviour, need small h

$\Rightarrow$ $n = \frac{T}{h} + 1$ (# steps in our method) is going to be huge!

So very slow to solve
(Also, small h not good for roundoff)

Ex Simulating Earth's climate.

One reason it's hard are the different timescales

- Temperature might change 3°C in an hour, but we need to simulate for 100,000 years!

The name stiff comes from mass-spring systems

Ex Linear ODEs: $y'' + 150y' + 5000y = \cos(t)$

represents a driven mass-spring ($m = 1\text{ kg}$, $b = 150$ is damping, $k = 5000$ is spring constant)

1) homogeneous ODE $y''+150y'+5000y=0$ has solutions of the form $e^{\lambda t}$. Plug this in to get

$$\lambda^2 + 150\lambda + 5000 = 0$$

$$(\lambda+50)(\lambda+100)=0 \implies \lambda \in \{-50,-100\}$$

$$y_{\text{homogeneous}}(t) = c_1 e^{-50t} + c_2 e^{-100t}$$ "transient" solution, "fast" timescale

2) particular solution

$$y_p = c_3 \cdot \cos(t) + c_4 \cdot \sin(t)$$ "Steady state" solution, "Slow" timescale

3) $$y = \underbrace{c_1 e^{-50t} + c_2 e^{-100t}}_{\text{fast}} + \underbrace{c_3\cos(t) + c_4\sin(t)}_{\text{slow}}$$

What to do about stiffness?

Avoid making h extremely small (and give up on capturing all the fast dynamics) if we have to

But some methods can't handle a large value of h ...

Absolute Stability

Previously, "stability" meant the root condition, equivalent (under consistency) to "0-stability"

i.e., $0 \le t \le T$, take $h \to 0$ ... hence the name "0-stability"

Now, focus on absolute stability

Fix $h>0$ (do not take $h\to 0$), solve on interval $0\le t\le T$

take $T \to \infty$.

To define this notion, we need to introduce:

Def (Dahlquist's) Test Equation

$$y' = \lambda y, \quad y(0) = 1$$

our book allows $y(0)=\alpha$ but you can take $\alpha=1$ wlog

Our book assumes $\lambda < 0$ (not all books do... it doesn't affect the math, but the interpretation makes the most sense if $\lambda<0$. But $\lambda \in \mathbb{C}$ can make sense too)

True solution is $y(t) = e^{\lambda t}$.

Why? Think of a 1st order linear ODE $\vec{y}' = A\cdot\vec{y}$ for a matrix A

Then if A is diagonalizable, after a change of basis, each component of $\vec{y}$, y_i, will satisfy $y_i' = \lambda_i y_i$

where λ_i is an eigenvalue of A.

If A has eigenvalues of very disparate magnitudes, it's a stiff problem.

Not all stiff ODEs are linear, but often linear approximations give us good rules-of-thumb

Let's solve the test equation using Euler's method w/ stepsize h

$$f(t,y) := \lambda \cdot y$$

$$w_0 = 1$$

$$w_1 = w_0 + h f(t_0, w_0)$$

$$= w_0 + h\lambda w_0$$

$$= (1+h\lambda) w_0$$

Similarly,

$$w_2 = (1+h\lambda) w_1 = (1+h\lambda)^2 w_0$$

So generally

$$w_n = \underbrace{(1+h\lambda)^n}_{\text{does this } \to 0 \text{ as } n\to\infty\,?} w_0, \quad \text{which approximates } y_n = y(t_n) = \underbrace{e^{nh\lambda}}_{\to 0 \text{ if } \lambda<0 \text{ as } n\to\infty}$$

$y(t) = e^{\lambda t}$

$t_n = n \cdot h$

$$\lim_{n\to\infty} a^n = \begin{cases} 1 & \text{if } a=1 \\ \text{DNE} & \text{if } |a|\geq 1 \text{ and } a\neq 1 \\ 0 & \text{if } |a|<1 \end{cases}$$

So want $|1+h\lambda| < 1$, i.e., $\boxed{h\lambda \text{ is in a ball centered at } -1, \text{ of radius } 1}$

Repeat using backward (implicit) Euler

$$w_0 = 1 \qquad f(t,y) = \lambda y \text{ as before}$$

$$\left.\begin{aligned} w_1 &= w_0 + h f(t_1, w_1) \\ &= w_0 + h\lambda w_1 \end{aligned}\right\} \text{solve implicit eq'n, } (1-h\lambda) w_1 = w_0$$

$$w_1 = \frac{1}{1-h\lambda} w_0$$

$$\ldots \quad w_n = \frac{1}{(1-h\lambda)^n} w_0.$$

Does $w_n \to 0$ as $n\to\infty$?

i.e., is $\left|\frac{1}{1-h\lambda}\right| < 1$? If $h\lambda$ is negative (makes sense... $\lambda<0$, $h>0$)

then $1-h\lambda > 1$

$$\Rightarrow \frac{1}{1-h\lambda} < 1 \quad \checkmark$$

What we've just done is find a region of stability: values of $h\lambda$ for which $\lim_{n\to\infty} w_n = 0$

(not $h \to 0$)

Def (part 1) For a one-step method $w_{i+1} = w_i + h\phi(t_i, w_i, h)$,

apply it to $f(t,y) = \lambda y$ and rewrite as $w_{i+1} = Q^{\text{1-step}}(h\lambda)\, w_i$.

Then the region R of absolute stability is

$$R = \{ h\lambda \in \mathbb{C} : |Q^{\text{1-step}}(h\lambda)| < 1 \}.$$

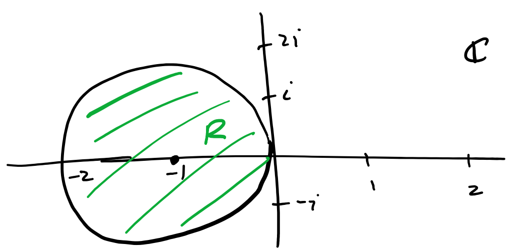
(why allow $\mathbb{C}$? $y(t) = e^{\lambda t}$

if λ is imaginary, this is a sin or cos

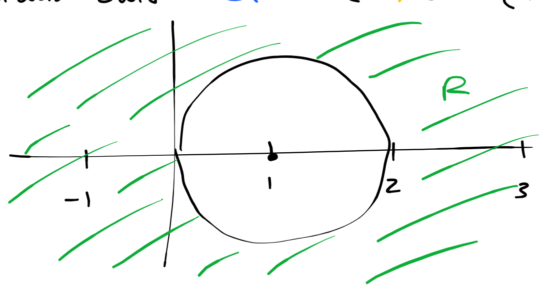
Euler's id: $e^{ix} = \cos(x) + i \cdot \sin(x)$)

So we've derived

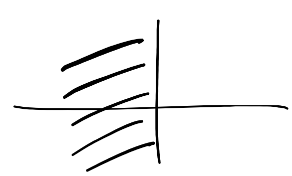
Forward Euler: $Q^{\text{1-step}}(h\lambda) = 1 + h\lambda$



Backward Euler: $Q^{\text{1-step}}(h\lambda) = (1 - h\lambda)^{-1}$



We call backward Euler A-Stable because R contains the entire left-half plane



Multistep methods

Same principle but shortcut to find region R of abs. stability.

$$w_{i+1} = a_{m-1} w_i + a_{m-2} w_{i-1} + \dots + a_0 w_{i+1-m}$$

$$+ h \cdot \left(b_m f_{i+1} + b_{m-1} \cdot f_i + \dots + b_0 \cdot f_{i+1-m} \right)$$

(general multi-step method)

$f_i = f(t_i, w_i)$. Apply to our <u>test eq'n</u> $y' = \lambda y$ so $f_i = f(t_i, w_i) = \lambda w_i$

to get

$$w_{i+1} = a_{m-1} w_i + a_{m-2} w_{i-1} + \dots + a_0 w_{i+1-m}$$

$$+ h \left(b_m \lambda w_{i+1} + b_{m-1} \lambda w_i + \dots + b_0 \lambda w_{i+1-m} \right)$$

so rewrite as

$$(1 - b_m h\lambda) w_{i+1} - (a_{m-1} + h\lambda b_{m-1}) w_i - \dots - (a_0 + h\lambda b_0) w_{i+1-m} = 0$$

This is a linear, homogeneous difference equation like we saw in ch 5.10

To solve it, guess $w_n = z^n$ for some $z \in \mathbb{C}$

$\Rightarrow$ z solves the polynomial equation

$$\underbrace{(1 - b_m h\lambda) z^m - (a_{m-1} + h\lambda b_{m-1}) z^{m-1} - \dots - (a_0 + h\lambda b_0) \cdot z^0}_{Q^{(h\lambda)}(z)} = 0$$

⚠ Confusingly, the book calls this Q the "characteristic polynomial" $\longrightarrow$ $Q^{(h\lambda)}(z)$

denotes that coefficients depend on $h\lambda$

but it's <u>not</u> the same "characteristic polynomial" as in ch 5.10

and it turns out that if all roots β_j of $Q^{(h\lambda)}$ are simple,* then all sol'n to the difference equation can be written as $w_n = \sum_{k=1}^{m} c_k \beta_k^{\,n}$

(c_k: coefficient)

So if we want to <u>guarantee</u> $\lim_{n\to\infty} w_n = 0$, we want all $|\beta_k| < 1$

* (if roots aren't simple it's more complicated, but end up with also we want all $|\beta_k| < 1$)

<u>Def</u> (part 2) The region R of absolute stability for a multistep method is

$$R = \{ h\lambda \in \mathbb{C} : \text{all roots } \beta \text{ of } Q^{(h\lambda)} \text{ satisfy } |\beta| < 1 \}$$

<u>Punchline</u>

Stiff eq'n often have a component like $e^{\lambda t}$ for $\lambda < 0$

The effect of this part in w_n should decay, so we want to pick h such that $h\lambda \in R$

In general, we prefer methods with a large R

... especially A-Stable methods

these are "unconditionally" stable: any $h>0$ is stable

Fact: implicit methods typically have larger regions of stability than comparable explicit methods

... so for stiff problems, we usually prefer implicit methods despite their extra hassle.

⚠ Finding roots of $Q^{(h\lambda)}$: λ is a parameter now, not a variable (the variable is z)

Examples

① Implicit Trapezoidal (= AM2)

$$w_{i+1} = w_i + \frac{h}{2}\left(f_{i+1} + f_i \right)$$

So $$Q^{(h\lambda)} = \left(1 - \tfrac{1}{2}h\lambda\right)\cdot z - \left(1 + \tfrac{1}{2}h\lambda\right)$$

$\Rightarrow$ only one root, $$\beta = \frac{1+\frac{1}{2}h\lambda}{1-\frac{1}{2}h\lambda} = \frac{2+h\lambda}{2-h\lambda}$$

So $h\lambda \in R$ when $\left| \frac{2+h\lambda}{2-h\lambda} \right| < 1$

i.e., $|2 + h\lambda| < |2 - \underbrace{h\lambda}_{\text{write as } a+ib\in\mathbb{C}}|$

i.e., $|(2+a)+ib|^2 < |(2-a)-ib|^2$

$$(2+a)^2 + \cancel{b^2} < (2-a)^2 + \cancel{b^2} \qquad b \text{ plays no role}$$

$$\cancel{2^2} + 4a + \cancel{a^2} < \cancel{2^2} - 4a + \cancel{a^2}$$

$$\left.\begin{array}{l} a < -a \\ 2a < 0 \end{array}\right\} \Rightarrow a < 0$$

So $R = \{ h\lambda \in \mathbb{C} : \operatorname{Re}(h\lambda) < 0 \}$ = left-half plane

$\Rightarrow$ AM2/trapezoidal is A-stable

So, Backward Euler and AM2/trapezoidal are A-stable

... it turns out there are no other A-stable multistep methods

("2nd Dahlquist Barrier": All A-stable linear multistep methods must be implicit and be $O(h^2)$ or less)

More examples at the .ipynb demos since involve a lot of plotting

Chapter Summary

- We always want consistent methods

- We like high-order local truncation error
- Explicit methods are simpler to work with, faster per iteration

- Want 0-stability / root condition

- Want large region of absolute stability especially for stiff problems

These are often at odds with each other:

Higher order usually means smaller region of absolute stability

Explicit usually means smaller region of absolute stability

- Multistage (RK) have good 0-stability, and easy to work w/ adaptive stepsize $\Rightarrow$ ode45 in Matlab is often your 1st choice

- Implicit multistep methods have good stability (and predictor-corrector helps w/ implicitness)

ode113

$\Rightarrow$ often your 1st choice for stiff eq'n

ode15s, ode23s